

Envelope Interfaces for Geoscientific Processing with High Performance Computing and Information Systems

Claus-Peter Rückemann

Leibniz Universität Hannover (LUH),

Westfälische Wilhelms-Universität Münster (WWU),

North-German Supercomputing Alliance (HLRN), Germany

Email: ruckema@uni-muenster.de

Abstract—The results presented in this paper describe the achievements with the development of techniques, implementing a standard way for computing communication in complex integrated information and computing systems. Compute Envelopes can provide means for generic data processing and flexible information exchange, namely computation objects. Targeting mission critical environments, the interfaces can embed instruction information, validation and verification methods. The application covers challenges of collaborative implementation, legal, and security issues with these processes. A major task is integrating information systems with Distributed and High Performance Computing (HPC) resources in natural sciences disciplines, scientific information systems, for building integrated public/commercial information system components within the e-Society. The main focus of this paper is on how to simplify the integration of information and computing resources from modular system architectures, using envelope techniques reducing complexity and strengthening trust aspects in future integrated information and computing systems.

Keywords—*Computing and Information Systems; Geoscientific Processing; Distributed and High Performance Computing.*

1. Introduction

There are two main objectives for interfacing modular complex integrated information and computing systems: “trust in computing” and “trust in information”. This paper concentrates on implementing methods for flexible use of envelope interfaces for use with integrated information and computing systems for managing objects and strengthen trust with systems from natural and geosciences, spatial sciences, and remote sensing as to be used for application, e.g., in environment management, healthcare or archaeology. Spatial means are tools, for the sciences involved. Therefore processing and computing is referred to the content which is embedded and used from the visual domain.

Over the last years a long-term project, Geo Exploration and Information (GEXI) [1] for analysing national and international case studies and creating as well as testing various implementation scenarios, has shown the two trust groups of systems, reflected by the collaboration matrices [2]. It has examined chances to overcome the deficits, built a collaboration framework and illuminated legal aspects and benefits [3]. The information and instructions handled within

these systems is one of the crucial points while systems are evolving by information-driven transformation [4].

For computing and information intensive systems the limiting constraints are manifold. Recycling of architecture native and application centric algorithms is very welcome. In order to reuse information about these tasks and jobs, it is necessary to enable users to separate the respective information for system and application components. This can be done by structured envelope-like descriptions containing essential workflow information, algorithms, instructions, data, and meta data. The container concept developed has been called Compute Envelope (CEN). The idea of envelope-like descriptive containers has been inspired by the good experiences with the concept of Self Contained applications (SFC) [5]. Envelopes can be used to integrate descriptive and generic processing information. Main questions regarding the topics of computing envelope interfaces are: Which content can be embedded or referenced in envelopes? How will these envelope objects be integrated into an information and computing system and how can the content be used? How can the context and environment be handled?

This paper is organised as follows. Section 2 presents motivation and implementation challenges. Section 3 describes the implementation complexity. Sections 4 and 5 show the basics of envelope files, architecture, and meta data aspects. Sections 6 and 7 describe the processes of compute access and report on the implementation for integrated systems. Sections 8, 9, and 10 present the evaluation and summarise the lessons learned, conclusion, and outlook.

2. Motivation

The complexity of integrated information and computing systems is an interesting factor that makes it well worth applying meta-instructions and signatures for algorithms and interfaces. Within these systems users will have some defined characteristics, so that these can be described in a flexible way. The economic profit is even greater in case of application with informations systems handling huge numbers of objects. In these cases, envelopes can be used to provide a standard interface for computing access.

The purpose of CEN files is to get recyclable algorithms

and information for defined system environments. If the algorithms handled for computing purposes would concentrate on ordinary processing batch scripts only, this would be pointless and tedious and anyone would need his own computing instructions. Today's information and computing systems are facing challenges from complex environments and heterogeneous content. These present aggravation conditions for any long term implementation.

3. Implementation complexity and amazements

There is a number of scenarios described in the previous publications [1], [2], [3], [6], [7], showing how “trust in computing” and “trust in information” can more easily be achieved by reducing complexity for development partners in otherwise very complex systems. Examples of elements for data objects being subject to handling and protection are:

- Vector data and multi-dimensional data,
- Raster data (aerial, remote sensing, and photographic),
- Primary and secondary spatial information,
- Calculation, measurement, and processing results,
- Meta data, instruction and interactive information,
- Commercially provided or licensed data, etc.

Most problems arise from complexity necessary to reflect the use cases and being built on prepackaged components each having own practical ‘amazements’ for integrated development and from content and context handling.

3.1. Problems for complex use cases

At today's state of development, available technology is still very limited if we take a look on long term issues, stability, and extendability. Integrated information and computing systems components make use of various technologies: batch systems, schedulers, IPC, sandboxing, embedded applications, browser plugins, remote execution, network and communication protocols, computing interfaces as well as public and sensitive data. The major motivation is to create an architecture of system components to ease the use of various content in different resources environmental context.

3.2. Content and context

The most important experience found from the case studies over the last years is, that besides the algorithmic content can be signed, the context of the CEN cannot be signed in any way. So, due to the complexity of the environment, the fitness for a specific purpose cannot be guaranteed.

What can be controlled is given in the following list. For some cases these characteristics can even be signed.

- Information and computing resources instructions,
- Links between the information and computing system,
- Prerequisites of the computing system,
- Processing directives and script elements,
- Input/output data necessary, etc.

What cannot be fully validated and ‘signed’ is environment and network specifications, nodes characteristics, state of the components of the system, . . .

The signer of a script would have to create and sign a compute envelope for a defined system context. This is comparable to signature for document files. These only refer to the content namely the object and not for the context outside, where the data will be used, namely the environment and action. The parts of the algorithm that cannot be signed, can in many cases be checked for logical and consistent state. The environment has to be validated and verified with the installation and configuration of the information and computing system. Looking at these aspects it would be beyond the scope of this paper as this will be future work.

4. Compute instructions into envelopes

SFC applications are used for over ten years now, but there has been no general concept for communication in integrated environments. These applications can hold data, executable binary and bytecode, components, configuration, libraries and other parts in a way supporting flexible and portable use. Even more flexibility can be achieved for non binary components and executables, meaning non system dependent elements. Starting from the proof of concept, a more suitable solution has now been created on a generic envelope base. An end-user public client application may be implemented via a browser plugin, based on appropriate services. The current solution is based on CEN files containing XML structures for handling and embedding compute information. Common envelopes that have been used with content signing, are Object Envelope (OEN) files. Envelope files can even be integrated with sources like GISIG Active Source, Active Map (GAS, GAM), with Object Oriented (OO) support, embedded into SFC applications. For various applications the use from within binary executables or TBC has been considered for any application part and data [5]. The case studies have shown good results as porting components can be made easier and application service providers and users can react faster on system modifications. Listing 1 shows a generic CEN file.

```

1 <ComputeEnvelope><!-- ComputeEnvelope (CEN)-->
2 <Instruction>
3 <Filename>Processing_Bat_GIS515.torque</Filename>
4 <Md5sum>...</Md5sum><Sha512sum>...</Sha512sum>
5 <DateCreated>2010-08-01:231523</DateCreated>
6 <DateModified>2010-08-01:232734</DateModified>
7 <ID>...</ID><CertificateID>...</CertificateID>
8 <Signature>...</Signature><Content>...</Content>
9 </Instruction>
10 </ComputeEnvelope>
    
```

Listing 1. Example for a Compute Envelope (CEN).

Listing 2 shows a small example using DataReference instead of embedded data inside a CEN.

```

1 ...<Content><DataReference>https://doi...</
  DataReference></Content>...
    
```

Listing 2. CEN using DataReference.

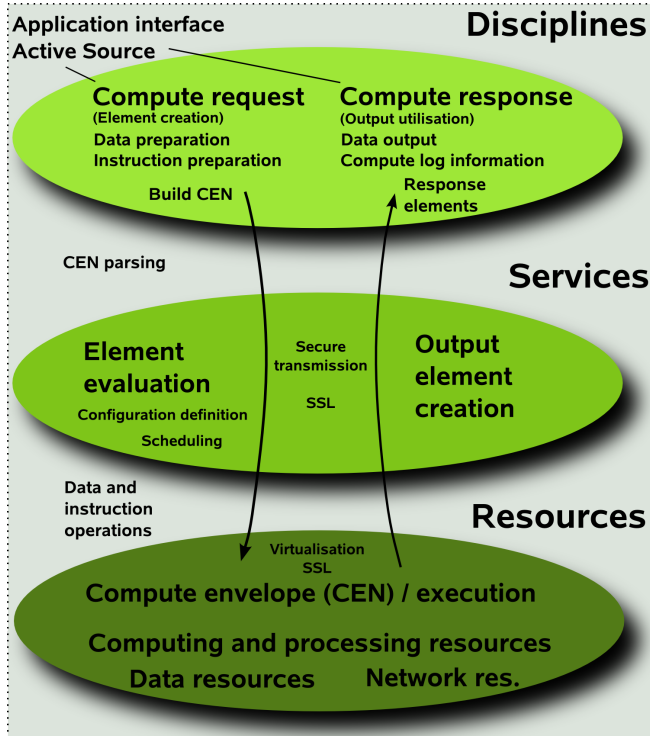


Figure 1. Compute access for integrated systems.

Content can be handled as content-stream or as content-reference (Listing 2). The way this will have to be implemented for different use cases depends on the situation, an in many cases on the size and number of data objects. One benefit of content-reference with high performant distributed or multicore resources is that references can be processed in parallel on these architectures. The number of physical parallel resources and the transfer capacities inside the network are limiting factors.

5. Fundamental implementation architecture

The fundamental architecture is based on a layered concept for the implementation and operation of information and computing systems [3]. Currently “trust in computing” can cover the content aspects. Context aspects are out of scope with today's systems. For the three development layers this mainly states tasks for services and resources layers.

Figure 1 shows the compute access for integrated information and computing systems. Providing application support for integrated systems does base on two assumptions, that applications can be modified to support an integration of compute resources and that the configuration of the system resources infrastructure can be configured in a suitable way. This means extending the application sources on one hand and to enable automating of access and workflow cycles. The meta structure can be used to store the instruction sets and to describe the job and the environment. Various meta data is necessary to describe the signed object data. For chronology as well as for plausibility, the security of the

time and data association is important. Integrated system components as well as interested parties must be able to use this meta data as well as means must be available to verify that the time stamps associated with an object are authentic and hold integrity (trusted time stamp authorities). This may be a function of the CA, respective a dedicated time server service. The envelope is able to describe any form of embedded or referred meta data.

6. Compute access for integrated systems

From the experiences of the case studies, the following passages describe the workflow (Figure 1) necessary to implement integrated information and computing systems in context with envelopes interfaces.

Based on the three layers of the Grid-GIS house framework, namely disciplines, services, and resources, the compute access is established by several steps, consisting of a compute request block and a compute response block.

6.1. The request process

Starting with the application interface a compute request starts with the element creation. The main parts consist of preparation of the data needed for computing or processing and the preparation of the necessary instructions. In the case studies Active Source applications have been used. From these prerequisites the CEN are built.

- **Disciplines layer:** The user's compute request is started based on computation data and the instruction information necessary for processing. Elements for the CEN are created by the user. The CEN is built from the elements, supported by application functions.
- **Services layer:** The elements are evaluated and adapted by the system configuration definition. The instruction sets are prepared for scheduling.
- **Resources layer:** Data and instruction operations are handled by batch system or interactive use. Compute, data, network, and storage resources are used with elements and configuration by services layer definition.

Depending on the system configuration, architecture, and allowed user behavior, the processing can run interactively or in batch mode. This can take from seconds to days for elements, depending on the purpose of the integrated system concept, before data will be suitable to be delivered for the response process.

6.2. The response process

With the request process delivering output, the following steps take place:

- **Resources layer:** The resulting output will be handled as described in the CEN instructions. Very large data can be stored on appropriate storage resources for later use, smaller or interactive data can be directly delivered to the services layer.

- **Services layer:** Services functions handle the output and do create output elements, delivered to the user or interface defined in the original CEN envelope.
- **Disciplines layer:** The data from the output elements is delivered for utilisation to the user or interface, e.g., to be interactively integrated into the application.

7. Implemented solution for integrated systems

For most interactive information system components a configuration of the distributed resources environment was needed. In opposite to OEN use, making it necessary to have referenced instead of embedded data for huge data sets, for CEN it should be possible to embed the essential instruction data in most cases. So there is less need for minimising data overhead and communication. Envelope technology is meant to be a generic extensible concept for information and computing system components. Figure 2 shows the workflow with application scenarios from the GEXI case studies. Future objectives for client components are:

- Channels for limiting communication traffic,
- Qualified signature services and accounting,
- Using signed objects without verification,
- Verify signed objects on demand.

The tests done for proof of concept have been in development stage. A more suitable solution has now been created on a generic envelope base. An end-user public client application may be implemented via a browser plugin, based on appropriate services. The current solution is based on CEN files containing XML structures for handling and embedding data and information.

7.1. Integrated components in practice

When taking a look onto different batch and scheduling environments one can see large differences in capabilities, handling different environments and architectures. In the last years experiences have been gained in handling simple features for different environments for High Throughput Computing like Condor [8], workload schedulers like LoadLeveler [9] and Grid Engine [10], and batch system environments like Moab/Torque [11], [12]. Batch and interactive features are integrated with Active Source event management [5]. Listing 3 shows a small example of a CEN embedded into an Active Source component.

```

1  #BCMT-----
2  ##EN \gisignip{Object Data: Country Mexico}
3  #ECMT-----
4  proc create_country_mexico {} {
5  global w
6  # Sonora
7  $w create polygon 0.938583i 0.354331i 2.055118i ...
8  #BCMT-----
9  ##EN \gisignip{Compute Data: Compute Envelope (CEN)}
10 #ECMT-----
11 #BCEN <ComputeEnvelope>
12 ##CEN <Instruction>
13 ##CEN <Filename>Processing_Bat_GIS515.torque</Filename>
14 ##CEN <Md5sum>...</Md5sum>
15 ##CEN <Sha1sum>...</Sha1sum>

```

```

16 ##CEN <Sha512sum>...</Sha512sum>
17 ##CEN <DateCreated>2010-09-12:230012</DateCreated>
18 ##CEN <DateModified>2010-09-12:235052</DateModified>
19 ##CEN <ID>...</ID><CertificateID>...</CertificateID>
20 ##CEN <Signature>...</Signature>
21 ##CEN <Content>...</Content>
22 ##CEN </Instruction>
23 #ECEN </ComputeEnvelope>
24 ...
25
26 proc create_country_mexico_autoevents {} {
27 global w
28 $w bind legend_infopoint <Any-Enter> {set killatleave [
   exec ./mexico_legend_infopoint_viewall.sh $op_parallel
 ] }
29 $w bind legend_infopoint <Any-Leave> {exec ./
   mexico_legend_infopoint_kaxv.sh }
30 $w bind tulum <Any-Enter> {set killatleave [exec
   $appl_image_viewer -geometry +800+400 ./
   mexico_site_name_tulum_temple.jpg $op_parallel ] }
31 $w bind tulum <Any-Leave> {exec kill -9 $killatleave }
   } ...

```

Listing 3. CEN embedded with Active Source.

Interactive applications based on Active Source have been used on Grid, Cluster, and HPC (MPP, SMP) systems [6].

7.2. Resources interface

Using CEN features, it is possible to implement resources access on base of validation, verification, and execution. The sources (Listing 4, 5) can be generated semi-automatically and called from a set of files or can be embedded into an actmap component, depending on the field of application.

```

1 <ComputeEnvelope><!-- ComputeEnvelope (CEN)-->
2 <Instruction>
3 <Filename>Processing_Batch_GIS612.pbs</Filename>
4 <Md5sum>...</Md5sum>
5 <Sha1sum>...</Sha1sum>
6 <Sha512sum>...</Sha512sum>
7 <DateCreated>2010-08-01:201057</DateCreated>
8 <DateModified>2010-08-01:211804</DateModified>
9 <ID>...</ID>
10 <CertificateID>...</CertificateID>
11 <Signature>...</Signature>
12 <Content><DataReference>https://doi...</DataReference><
   /Content>
13 <Script><Pbs>
14 <Shell>#!/bin/bash</Shell>
15 <JobName>#PBS -N myjob</JobName>
16 <Oe>#PBS -j oe</Oe>
17 <Walltime>#PBS -l walltime=00:10:00</Walltime>
18 <NodesPpn>#PBS -l nodes=8:ppn=4</NodesPpn>
19 <Feature>#PBS -l feature=ice</Feature>
20 <Partition>#PBS -l partition=hannover</Partition>
21 <Accesspolicy>#PBS -l naccesspolicy=singlejob</
   Accesspolicy>
22 <Module>module load mpt</Module>
23 <Cd>cd $PBS_O_WORKDIR</Cd>
24 <Np>np=$(cat $PBS_NODEFILE | wc -l)</Np>
25 <Exec>mpieexec_mpt -np $np ./dyna.out 2>&1</Exec>
26 </Pbs></Script>
27 </Instruction>
28 </ComputeEnvelope>

```

Listing 4. Embedded Active Source MPI script.

Examples for using High Performance Computing and Grid Computing resources include batch system interfaces and job handling. Job scripts from this type will on demand (event binding) be sent to the batch system for processing. The Actmap Computing Resources Interface (CRI) is an example for an actmap library (actlcric) containing functions and

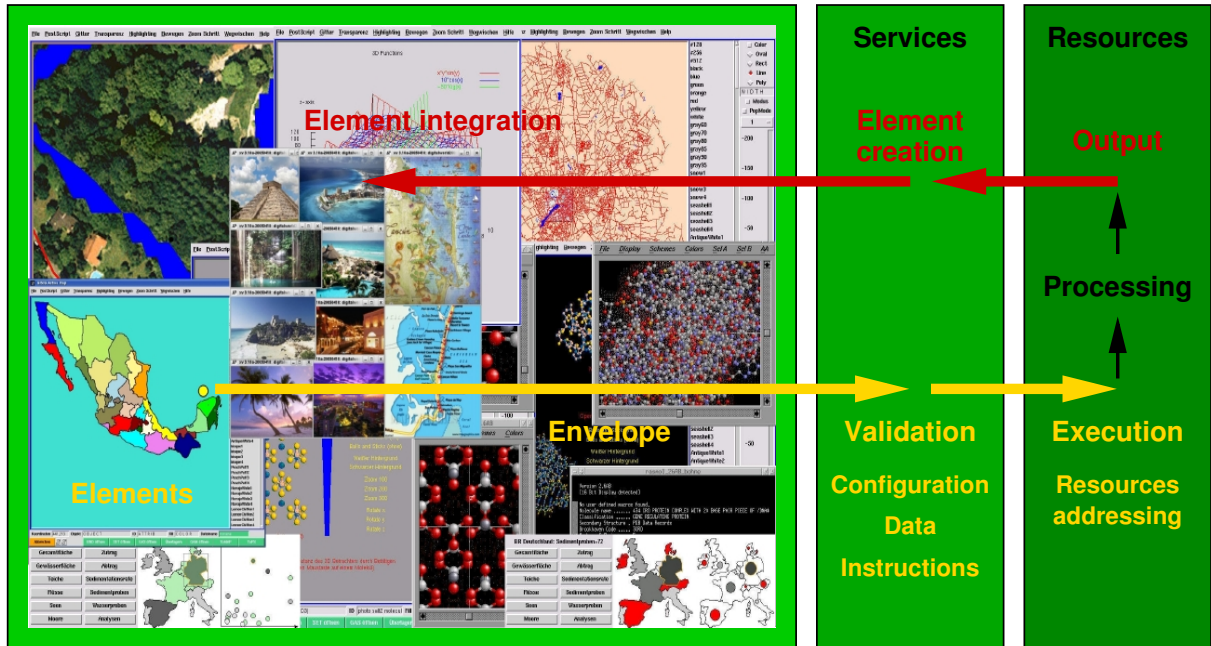


Figure 2. Workflow with application scenarios from the GEXI case studies.

procedures and even platform specific parts in a portable way. CRI can be used for handling computing resources, loading Tcl or TBC dynamically into the stack [13] when given set behaviour_loadlib_actlib "yes".

```

1 <ComputeEnvelope><!-- ComputeEnvelope (CEN)-->
2 <Instruction>
3 <Filename>Processing_Batch_GIS612.pbs</Filename>
4 <Md5sum>...</Md5sum>
5 <Shalsum>...</Shalsum>
6 <Sha512sum>...</Sha512sum>
7 <DateCreated>2010-08-01:201057</DateCreated>
8 <DateModified>2010-08-01:211804</DateModified>
9 <ID>...</ID>
10 <CertificateID>...</CertificateID>
11 <Signature>...</Signature>
12 <Content><DataReference>https://doi...</DataReference><
  /Content>
13 <Script><Condor>
14 <Environment>universe = standard</Environment>
15 <Exec>executable = /home/cpr/grid/job.exe</Exec>
16 <TransferFiles>should_transfer_files = YES</
  TransferFiles>
17 <TransferInputFiles>transfer_input_files = job.exe,job.
  input</TransferInputFiles>
18 <Input>input = job.input</Input>
19 <Output>output = job.output</Output>
20 <Error>error = job.error</Error>
21 <Log>log = job.log</Log>
22 <NotifyMail>notify_user = ruckema@uni-muenster.de</
  NotifyMail>
23 <Requirements>
24 requirements = (Memory >= 50)
25 requirements = ( ( OpSys=="Linux" ) || (OpSys=="AIX" ) ) && (
  Memory >= 500 )
26 </Requirements>
27 <Action>queue</Action>
28 </Condor></Script>
29 </Instruction>
30 </ComputeEnvelope>
    
```

Listing 5. Embedded Active Source Condor script.

With Actmap CRI being part of Active Source, calls to parallel processing interfaces, e.g., using InfiniBand, can

be used, for example MPI (Message Passing Interface) and OpenMP, already described for standalone job scripts for this purpose, working analogical. [7].

8. Evaluation

Targets for handling computing content were integrated information and computing systems. It has been possible to describe and handle components from prepackaged software, interfaces, and architectures within the envelopes for generic processing of geo- and related data. The primary benefits of the solution using CEN as stated from the case studies are:

- Build a defined interface between dedicated information system components and computing system components.
- Unique algorithm for using environment components.
- Integration of information and computing systems.
- Speed-up the development of new and modification of existing components in complex environments.
- Portable, transparent, extendable, flexible, and scalable.
- Hierarchical structured meta data, easily parsable.
- OO-support (object, element) on application level.
- Multi-system support.
- Support for validation and verification of object elements via Public Key Infrastructure (PKI).
- Usable with sources and binaries like Active Source.

Main drawbacks are:

- Additional layer.
- Complexity of parsing and configuration.

The context is an important aspect, though it cannot be called “drawback” here. With closed products, e.g., when memory requirements are not transparent, it is difficult for users to specify their needs. Anyhow, testing is in many cases not

the answer in productive environments. Separate measures have to be taken to otherwise minimise possible problems and ease the use of resources in productive operation.

Even in the face of the drawbacks, for information systems making standardised use of large numbers of accesses via the means of interfaces, the envelopes can provide efficient management and access, as programming interfaces can.

9. Lessons learned

CEN based on generic envelopes has provided a very flexible and extensible solution for creating portable, secure processing components with integrated information and computing systems. The case study showed that nearly any instruction set and data structure can be handled with CEN in embedded or referred use. Instruction information, meta data, signatures, and check sums can be used and customised in various ways for implementing information and computing system components. Support for transfer and staging of data in many aspects further depends on system configuration and resources as for example physical bottlenecks cannot be eliminated by any kind of software means. For future integrated information and computing systems an interface layer between user configuration and system configuration would be very helpful. From system side in the future we need working and least operation-invasive operating system resources limits, e.g., for memory and a flexible limits management.

10. Conclusion and future work

The envelopes have been successfully used for various Active Source components, for all data objects in use, embedded and referenced. Objects can be integrated in various native ways so the content is available as 'standard' content. As the context of the system environment existing, cannot be comprehensively defined, this leads to some interesting consequences. Exchange between systems is quite expendable without common standards. What we would need for future systems is some virtualised processing sandbox for integrating batch and scheduling systems, providing standardised definable interfaces. This concept has demonstrated a basic approach in order to begin to pave the way for information and computing systems and show the next aspects to go on with.

Acknowledgements

We are grateful to all national and international academic and industry partners in the GEXI cooperations for the innovative constructive work and to the colleagues at the Leibniz Universität Hannover, IRI, HLRN, WWU, ZIV, D-Grid for participating in fruitful case study examples and the participants of the EULISP Programme for prolific discussion of scientific, legal, and technical aspects over the last years.

References

- [1] "Geo Exploration and Information (GEXI)," 1996, 1999, 2010, URL: <http://www.user.uni-hannover.de/cpr/x/rprojs/en/index.html#GEXI> (Information) [accessed: 2010-05-02].
- [2] C.-P. Rückemann, "Legal Base for High End Computing and Information System Collaboration and Security," *International Journal on Advances in Security*, vol. 3, no. 3&4, 2010, (to appear), ISSN: 1942-2636, URL: <http://www.iariajournals.org/security/> [acc.: 2010-10-11].
- [3] C.-P. Rückemann, "Legal Issues Regarding Distributed and High Performance Computing in Geosciences and Exploration," in *Proceedings of the Int. Conf. on Digital Society (ICDS 2010), The Int. Conf. on Technical and Legal Aspects of the e-Society (CYBERLAWS 2010), February 10-16, 2010, St. Maarten, Netherlands Antilles*. IEEE Computer Society Press, IEEE Xplore Digital Library, 2010, pp. 339-344, ISBN: 978-0-7695-3953-9, URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5432414> [acc.: 2010-03-28].
- [4] M. Mackert, P. Whitten, and B. Holtz, *Health Infonomics: Intelligent Applications of Information Technology*. IGI Global, 2009, pp. 217-232, chapter XII, in: Pankowska, M. (ed.), *Infonomics for Distributed Business and Decision-Making Environments: Creating Information System Ecology*, ISBN: 1-60566-890-7, DOI: 10.4018/978-1-60566-890-1.ch010.
- [5] C.-P. Rückemann, "Beitrag zur Realisierung portabler Komponenten für Geoinformationssysteme. Ein Konzept zur ereignisgesteuerten und dynamischen Visualisierung und Aufbereitung geowissenschaftlicher Daten," Diss., Westfälische Wilhelms-Universität, Münster, Deutschland, 2001, 161(xxii+139)S., URL: <http://wwwmath.uni-muenster.de/cs/u/ruckema/x/download/dis3acro.pdf> [accessed: 2009-11-16].
- [6] C.-P. Rückemann, "Geographic Grid-Computing and HPC empowering Dynamical Visualisation for Geoscientific Information Systems," in *Proceedings of the 4th International Conference on Grid Service Engineering and Management (GSEM), September 25-26, 2007, Leipzig, Deutschland, co-located with Software, Agents and services for Business, Research, and E-sciences (SABRE 2007)*, R. Kowalczyk, Ed., vol. 117. GI-Edition, Lecture Notes in Informatics (LNI), Gesellschaft für Informatik e.V. (GI), 2007, pp. 66-80, ISBN: 78-3-8579-211-6, ISSN: 1617-5468.
- [7] C.-P. Rückemann, "Dynamical Parallel Applications on Distributed and HPC Systems," *International Journal on Advances in Software*, vol. 2, no. 2, 2009, ISSN: 1942-2628, URL: <http://www.iariajournals.org/software/> [accessed: 2009-11-16].
- [8] "Condor, High Throughput Computing," 2010, URL: <http://www.cs.wisc.edu/condor/> [accessed: 2010-10-10].
- [9] "IBM Tivoli Workload Scheduler LoadLeveler," 2005, URL: <http://www-03.ibm.com/systems/software/loadleveler/> [accessed: 2010-10-10].
- [10] "Sun Grid Engine," 2010, URL: <http://gridengine.sunsource.net/> [accessed: 2010-10-10].
- [11] "Moab: Admin Manual, Users Guide," 2010, URL: <http://www.clusterresources.com> [acc.: 2010-10-10].
- [12] "Torque Admin Manual," 2010, URL: <http://www.clusterresources.com/torquedocs21/> [accessed: 2010-10-10].
- [13] "Tcl Developer Site," 2010, URL: <http://dev.scripatics.com/> [accessed: 2010-10-10].